Computability and Complexity
Sample Exam Questions

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

Family Name:

Given Name:

Matriculation Number:

Study Code:

Total: 100 Points.

$\geq$ 51 Points: GEN4
$\geq$ 64 Points: BEF3
$\geq$ 77 Points: GUT2
$\geq$ 89 Points: SGT1

Note: these questions amount to substantially more than 100 points.

1. (20P) Let L be the language over the alphabet $\{0, 1\}$ whose words contain the string 10, but not at the beginning of the word.
   a) (4P) Give a regular expression that denotes L.
   b) (6P) Define a non-deterministic finite state machine $M = (Q, \Sigma, \delta, S, F)$ whose language is L (the transition function by both a table and a graph).
   c) (6P) Define a deterministic finite state machine whose language is L.
   d) (4P) Define a finite state machine whose language is the complement of L.
2. (10P) Construct a non-deterministic finite state machine whose language is denoted by the regular expression $1 + (2^* + 1 \cdot 2^* \cdot 3 \cdot (2 + 3)^*)^*$.
3. (16P) Which of the following languages over the alphabet $\{0, 1\}$ are regular? Justify your answers in detail.
   a) (4P) $L_a := \{0^m : m \in \mathbb{N} \wedge 2|m \wedge 3|m\}$
   b) (4P) $L_b := \{0^m 1^n : m, n \in \mathbb{N} \wedge 2|m \wedge 3|n\}$
   c) (4P) $L_c := \{0^m 1^n : m, n \in \mathbb{N} \wedge m|n\}$
   d) (4P) $L_d := \{0^m 1^n : m, n \in \mathbb{N} \wedge m|n \wedge 0 < n < 1000\}$
4. (16P) Let $\langle M \rangle$ denote the code of a Turing machine M with alphabet $\{0, 1\}$. Which of the following languages are recursively enumerable and/or recursive? Justify your answers in detail.
   a) (4P) $L_1 := \{\langle M \rangle : 10101 \in L(M)\}$
   b) (4P) $L_2 := \{\langle M \rangle : 10101 \notin L(M)\}$
   c) (4P) $L_3 := \{\langle M \rangle : L(M)$ ist recursively enumerable$\}$
   d) (4P) $L_4 := \{\langle M \rangle : L(M)$ ist recursive$\}$
5. (15P) Are the following statements true or not? Justify your answers in detail.
   a) (5P) If L_1 and L_2 are recursive languages, then also their difference $L_1 \backslash L_2$ is recursive.
   b) (5P) If L_1 is recursively enumerable and L_2 is recursive, then their difference is recursively enumerable.
   c) (5P) If L_1 and L_2 are recursively enumerable, then also their difference is recursively enumerable.

6. (25P) Are these statements true or not? Justify your answers in detail.

   Let L be the set of strings of form '$1^n + 1^m = 1^{n+m}$' (e.g. "111+11=11111" is in L).

   a) (5P) There exists a regular expression R with $L(R) = L$.
   b) (5P) There exists a grammar G with $L(G) = L$.
   c) (5P) There exists a Turing machine M with $L(M) = L$.
   d) (5P) L can be generated by a Turing machine.
   e) (5P) L is recursive.

7. (15P) Construct a LOOP program which computes the function $s(n) := \sum_{i=1}^{n} i$.

   Furthermore, give a primitive recursive definition of s or argue why this is not possible.

8. (12P) Are these statements true or not? Justify your answers in detail.

   a) (4P) For every primitive-recursive function f,

   $$t(n) := \min i : f(i) = n$$

   is μ-recursive; t is even primitive recursive.

   b) (4P) For every primitive-recursive function f, the function

   $$t(n,m) := \min i : n \leq i < m \wedge f(i) \neq 0$$

   is primitive recursive.

   c) (4P) For every primitive-recursive function f, the function

   $$t(n,m) := \begin{cases} m & \text{if } \forall i : n \leq i < m \Rightarrow f(i) = 0 \\ \min i : n \leq i < m \wedge f(i) \neq 0 & \text{else} \end{cases}$$

   is primitive recursive.

9. (10P) Is the following problem semi-decidable by a Turing machine? If yes, give an informal construction of this machine (pseudo-code and/or diagram plus explanation). If not, then justify your answer in detail.

   Decide for given Turing machine codes $\langle M_1 \rangle$ and $\langle M_2 \rangle$, whether $L(M_1) \cap L(M_2) \neq \emptyset$.

   Answer the question also for the problem $L(M_1) \cap L(M_2) = \emptyset$.

10. (10P) Formally prove or disprove $5n^2 + 7 = O(2^n)$.

11. (10P) Formally prove $\sum_{k=1}^{n} k^2 = n \cdot (n+1) \cdot (2n+1)/6$.

12. (10P) Formally prove that for all $n = 2^m$, the recurrence $T(1) = 1, T(n) = 4 \cdot T(n/2)$ is solved by $T(n) = n^2$.

13. (15P) Let $T(n)$ be the number of times that the command C is executed in the following program.

```
for (i=0; i<n; ++)
  for (j=i+1; j<n; j++)
    for (k=i; k<j; k++)
      C;
```

a) Compute $T(4)$.

b) Give an explicit definition of $T(n)$ by a nested sum and derive from this an asymptotic estimation $T(n) = \Theta(\ldots)$.

c) Give an explicit definition of $T(n)$ with the help of a single summation symbol (hint: first consider which pairs of i, j may arise in the program, then consider how often the innermost loop is executed for each pair, then consider how often the same iteration number for the innermost loop occurs for all pairs).

d) Give an explicit definition of $T(n)$ without using a summation symbol.

14. (10P) Consider two programs with the following shape

```
P1(a, b, ...):
  n = b-a;
  for (i=0; i<n; i++)
    for (j=i; j<n; j++)
      ...
  return ...
```

```
P2(a, b, ...):
n = b-a;
if (n <= 0) return ...;
for (i=0; i<7; i++) {
  c = ...i...;
  ...; P2(c, c+n/3, ...); ...;
}
return ...
```

where the parts marked as "..." are executed in time $O(1)$.

Which program runs faster for large input measures? Justify your answer in detail.

15. (25P) Take the function

```
static void P(int[] a, int i) {
  if (i == a.length) {
    System.out.println(Arrays.toString(a));
    return;
  }
  int t = a[i];
  for (int j=i; j<a.length; j++) {
    a[i] = a[j]; a[j] = t;
    P(a, i+1);
    a[j] = a[i];
```

```
  }
  a[i] = t;
}
```

We are interested in the number $T(n)$ of (recursive) invocations of P arising from a call of $P(a, 0)$ for an array a of length n.

a) (5P) Sketch a recursion tree for $P(a, 0)$ for $n = 4$. What is the number of nodes in each level of the tree?

b) (5P) Give a recurrence for $T(n)$.

c) (5P) Give the result of $T(n)$ as a summation.

d) (10P) Does $T(n) = O(2^n)$ hold? Does $T(n) = O(n^n)$ hold? Justify your answers in detail.

16. (28P) Are these statements true or not? Justify your answers in detail.

a) (4P) If both $f : \{0\}^* \to \{0\}^*$ and $g : \{0\}^* \to \{0\}^*$ are Turing-computable in polynomial time, then also $f \circ g$ is.

b) (4P) If there is a problem in $\mathcal{NP}$ that can be also solved deterministically in polynomial time, then $\mathcal{P} = \mathcal{NP}$.

c) (4P) If $\mathcal{P} \cap \mathcal{NPC} \neq \emptyset$, then $\mathcal{P} = \mathcal{NP}$.

d) (4P) If a problem P is decidable by a deterministic Turing machine, then also its complement is.

e) (4P) If a problem P is decidable by a deterministic Turing machine in polynomial time, then also its complement is.

f) (4P) If a problem P is decidable by a nondeterministic Turing machine, then also its complement is.

g) (4P) If a problem P is decidable by a nondeterministic Turing machine in polynomial time, then also its complement is.